Co-teaching guide

The AS and A-level Computer Science specifications have been written in such a way as to allow them to be taught at the same time. The key features of the specifications include:
the content of the AS specification is a subset of A-level specification content:
the A-level content goes into greater depth in some areas and so elements of the A-level/AS content will need to be revised at the end of the course of study
the software development section appears only at AS, however in co-teaching this section A-level candidates can gain useful insight into the processes required for non-exam assessment (NEA).
both the AS and A-level specifications have topics which are fundamental to Computer Science and can be taught together in the first year of the A-level course
topic numbering is consistent wherever possible to enable connections to be made in a straightforward way
similar assessment structures at both AS and A-level (with the exception of NEA at A-level) will allow the previous year's AS examinations to be used in centres as a progress check.
The main focus for the AS specification and the first year of the A-level specification is the teaching of a programming language and an understanding of the mechanics of programming. Both AS and A-level students will need time for these skills to develop over their respective courses.

Core programming skills

The skills required for the AS and A-level specifications are very similar when introducing programming to students. The A-level extends the skills required from AS by adding an appreciation of recursion and a requirement to explore object-oriented programming. The A-level also considers more advanced abstract data types such as graphs and trees. Object-oriented programming and abstract data types could be assessed in Paper 1 as part of an examined programming task or through more theoretical questions.

Centres familiar with the previous GCE Computing specification should see the additional A-level skills as an extension to those taught for programming. Any current schemes of work or lesson plans for teaching programming would be appropriate for the first year of this new specification.

Theory topics

The AS and A-level specifications have similar theory topics, but:
software development in AS is covered as part of the non-exam assessment (NEA) in A-level
in A-level additional topics are:
fundamentals of algorithms
fundamentals of databases
big data
fundamentals of functional programming.

At A-level, some of the AS topics are taken further and extended to increase students' depth of knowledge of the subject (see **appendix A**).
It is possible to co-teach a class of AS and first year A-level students the theory topics. This might mean that rather than finishing some topics off completely for the A-level you would revisit them in the second year.

The core topics to be taught in the first year are very similar to the topics covered in the first year of the previous specification. These topics include

data types
core programming concepts (declarations, sequence, iteration and selection)
subroutines
abstraction and automation
finite state machines
number systems
number bases (decimal, binary and hexadecimal)
binary number system (unsigned, signed two's complement and fixed point)
character coding systems (ASCII and Unicode)
representing images and sounds
data compression
encryption
hardware and software
programming language classification and translation
logic gates and Boolean algebra
internal hardware components
stored program concept and fetch–execute cycle
processor instruction set (assembly code)
consequences of uses of computing
communication and networking.

Possible co-teaching structure

Scenario	Winter term	Spring term	Summer term	Additional guidance
AS	Programming skills Fundamentals of programming Fundamentals of data structures Theory of computation Fundamentals of computer systems	Programming skills Skeleton program Fundaments of data representation Fundamentals of computer organisation and architecture Consequences of uses of computing Fundamentals of communication and networking	Software development Revision for AS exams Additional work on Skeleton program	
A-level	Programming skills Fundamentals of programming Fundamentals of data structures Theory of computation	Programming skills Fundamentals of data representation Fundamentals of computer organisation and architecture Consequences of uses of computing	Software development to enable students to commence NEA Commence work on A-level topics eg Object-oriented programming, and databases that are most likely to be useful for NEA	AS examinations could be used as mock papers / progress check for year 13

Scenario	Winter term	Spring term	Summer term	Additional guidance
	Fundamentals of computer systems	Fundamentals of communication and networking		

Appendix A - Table 1

Table 1 summarises the topics in both the AS and A-level specifications. Numbering for AS units begins with 3, the corresponding A-level numbering for units begins with 4.

AS topics which are not extended at A-level	Topics which are introduced in the AS and extended in A-level	A-level topics which are not included in the AS specification
	3.1.1.1 Data types (pointer and reference types included at A-level)	
3.1.1.2 Programming concepts		
3.1.1.3 Arithmetic operations in a programming language		
3.1.1.4 Relational operations in a programming language		
3.1.1.5 Boolean operations in a programming language		
3.1.1.6 Constants and variables in a programming language		
3.1.1.7 String-handling functions in a programming language		
3.1.1.8 Random number generation in a programming language		
3.1.1.9 Exception handling		
3.1.1.10 Subroutines (procedures/functions)		
3.1.1.11 Parameters of subroutines		
3.1.1.12 Returning a value/values from a subroutine		
3.1.1.13 Local variables in subroutines		
3.1.1.14 Global variables in a programming language		
		4.1.1.15 Role of stack frames in subroutine calls
		4.1.1.16 Recursive techniques
		4.1.2.1 Programming paradigms

AS topics which are not extended at A-level	Topics which are introduced in the AS and extended in A-level	A-level topics which are not included in the AS specification
		4.1.2.2 Procedural-oriented programming
		4.1.2.3 Object-oriented programming
3.2.1.1 Data structures		
3.2.1.2 One- and multi-dimensional arrays (or equivalent)		
3.2.1.3 Fields, records and files		
		4.2.1.4 Abstract data types/data structures
		4.2.2 Queues
		4.2.3 Stacks
		4.2.4 Graphs
		4.2.5 Trees
		4.2.6 Hash tables
		4.2.7 Dictionaries
		4.2.8 Vectors
	3.3.1.1 Analysis (A level includes requirement for interaction with the users and awareness that clarifying requirements may involve prototyping/agile approach)	
	3.3.1.2 Design (A level includes awareness that design can be an iterative process involving prototyping/agile approach)	
	3.3.1.3 Implementation (A level includes awareness that the solution may be arrived at using an iterative process with a focus on solving the critical path first)	
	3.3.1.4 Testing (A level includes knowledge of acceptance testing)	
3.3.1.5 Evaluation		
		4.3.1.1 Simple graph-traversal algorithms
		4.3.2.1 Simple tree-traversal algorithms
		4.3.3.1 Reverse Polish – infix transformations
		4.3.4.1 Linear search
		4.3.4.2 Binary search
		4.3.4.3 Binary tree search
		4.3.5.1 Bubble sort

AS topics which are not extended at A-level	Topics which are introduced in the AS and extended in A-level	A-level topics which are not included in the AS specification
		4.3.5.2 Merge sort
		4.3.6.1 Dijkstra’s shortest path algorithm
3.4.1.1 Problem solving		
3.4.1.2 Following and writing algorithms		
3.4.1.3 Abstraction		
3.4.1.4 Information hiding		
3.4.1.5 Procedural abstraction		
3.4.1.6 Functional abstraction		
3.4.1.7 Data abstraction		
3.4.1.8 Problem abstraction/reduction		
3.4.1.9 Decomposition		
3.4.1.10 Composition		
3.4.1.11 Automation		
	3.4.2.1 Finite state machines without output (with output required at A-level)	
		4.4.2.2 Maths for regular expressions
		4.4.2.3 Regular expressions
		4.4.2.4 Regular language
		4.4.3.1 Backus-Naur Form / syntax diagrams
		4.4.4.1 Comparing algorithms
		4.4.4.2 Maths for understanding the Big-O notation
		4.4.4.3 Order of complexity
		4.4.4.4 Limits of computation
		4.4.4.5 Classification of algorithmic problems
		4.4.4.6 Computable and non-computable problems
		4.4.4.7 Halting problem
		4.4.5.1 Turing Machine
3.5.1.1 Natural numbers		

AS topics which are not extended at A-level	Topics which are introduced in the AS and extended in A-level	A-level topics which are not included in the AS specification
3.5.1.2 Integer numbers		
3.5.1.3 Rational numbers		
3.5.1.4 Irrational numbers		
3.5.1.5 Real numbers		
3.5.1.6 Ordinal numbers		
3.5.1.7 Counting and measurement		
3.5.2.1 Number base		
3.5.3.1 Bits and bytes		
3.5.3.2 Units		
3.5.4.1 Unsigned binary		
3.5.4.2 Unsigned binary arithmetic		
3.5.4.3 Signed binary using two's complement		
	3.5.4.4 Numbers with a fractional part (floating point representation included at A-level, only fixed point at AS)	
		4.5.4.5 Rounding errors
		4.5.4.6 Absolute and relative errors
		4.5.4.7 Range and precision
		4.5.4.8 Normalisation of floating point form
		4.5.4.9 Underflow and overflow
3.5.5.1 Character form of a decimal digit		
3.5.5.2 ASCII and Unicode		
	3.5.5.3 Error checking and correction (Check sums included at A level)	
3.5.6.1 Bit patterns and images, sound and other data		
3.5.6.2 Analogue and digital		
	3.5.6.3 Analogue/digital conversion (uses included at A level)	
3.5.6.4 Bitmapped graphics		
		4.5.6.5 Vector graphics
		4.5.6.6 Vector graphics versus bitmapped graphics

AS topics which are not extended at A-level	Topics which are introduced in the AS and extended in A-level	A-level topics which are not included in the AS specification
3.5.6.5(AS)/3.5.6.7(A-level) Digital representation of sound		
3.5.6.6 (AS)/3.5.6.8 (A-level) MIDI		
3.5.6.7 (AS) / 3.5.6.9. (A-level) Data compression		
3.5.6.8(AS) / 3.5.6.10 (A-level) Encryption		
3.6.1.1 Relationship between hardware and software		
3.6.1.2. Classification of software		
3.6.1.3 System software		
3.6.1.4 Role of operating system		
3.6.2.1 Classification of programming languages		
3.6.3.1 Types of program translator		
	3.6.4.1 Logic gates (half-adder, adder and D-type flip-flops included in A-level)	
3.6.5 Boolean algebra		
3.7.1.1 Internal hardware components of a computer		
3.7.2.1 The meaning of the stored program concept		
3.7.3.1 The processor and its components		
3.7.3.2 The fetch–execute cycle and the role of registers within it		
3.7.3.3 The processor instruction set		
3.7.3.4 Addressing modes		
3.7.3.5 Machine-code/assembly language operations		
		4.7.3.6 Interrupts
3.7.3.6 (AS) / 4.7.3.7 (A-level) Factors affecting processor performance		
3.7.4.1 Input and output devices		
3.7.4.2 Secondary storage devices		
3.8 Consequences of uses of computing		

AS topics which are not extended at A-level	Topics which are introduced in the AS and extended in A-level	A-level topics which are not included in the AS specification
3.9.1.1 Communication methods		
3.9.1.2 Communication basics		
3.9.2.1 Network topology		
3.9.2.2 Types of networking between hosts		
3.9.2.3 Wireless networking		